HYBRID ALGORITHM WITH ADAPTIVE COMPLEXITY FOR INTEGER PEL MOTION ESTIMATION OF H.264

Li Zhang[1,2], Wen Gao[1,2]

[1]Institute of computing technology, Chinese Academy of Science
[2] Graduate school of Chinese Academy of science
{zhangli,wgao}@jdl.ac.cn

ABSTRACT

Owing to the great computation complexity of the ME (motion estimation) in video coding, a lot of fast ME algorithms have been proposed in literature. Most of them are designed based on a given hypothesis about the character of the video's motion field. But due to the inherent variety of the real-world video, these algorithms are not generally efficient, especially for the variable block size ME in H.264, only using one single ME strategy is difficult to get satisfactory result. This paper firstly will give two different fast ME algorithms, which aim at video sequences with different character. An adaptive strategy is proposed to combine the two algorithms together for H.264. Experimental result shows that this strategy can keep coding efficiency in all kinds of video sequences, meanwhile the computation complexity is reduced adaptively according to the video's content.

1. INTRODUCTION

The ME (motion estimation) is the most computationally expensive part of a video codec, especially for H.264 which has taken variable block size ME, the integer pel ME takes the heaviest computation burden. Many fast ME algorithms have been proposed ([1]-[3]) in literature. These traditional algorithms begin at an initial search point based on the MV (motion vector) prediction, the initial search point will be the center of the search window. These algorithms all think that the nearer a candidate position is to the center, the more likely it is to be the best candidate. Based on this center-biased hypothesis, these algorithms pay more attention to the central region of the search window. But this hypothesis is not always true, Following are two examples:

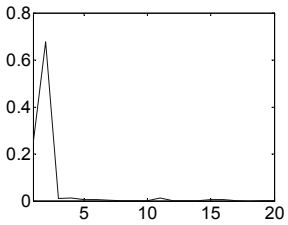

Figure 1-a Distribution of MV in sequence "Mobile and Calendar" (from frame 0 to frame 14)

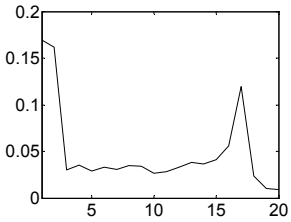

Figure 1-b Distribution of MV in sequence "Stefan" (from frame 240 to 259)

In the above two figures, the horizontal axis is the Euclidean distance between the final MV and the initial search point, the block matching size in the experiment is 16x16. We can see that the distribution of MV is highly center-biased in the clip of sequence "Mobile and Calendar". As for the clip of "Stefan" sequence which has very strong motion, the distribution of MV is relatively complex. We can imagine that the traditional center-biased algorithms can work well in the prior, but are easy to be trapped into local minimal in the later.

Noticing the drawback of the center-biased algorithms, researchers now have proposed fast ME algorithm aiming at the sequences with strong motions. For example, the UMHS (Unsymmetrical-cross Multi-Hexagon-grid Search) algorithm in [4] has been designed for the motion estimation of sequences with strong motion. It takes hybrid search patterns and

0-7803-9332-5/05/$20.00 ©2005 IEEE

multiple search steps to avoid the affect of local minimal, thus can maintain coding efficiency in the sequences with strong motion. Of course, the complexity of these algorithms is higher than the center-biased algorithms. Substantial computation waste will occur in the low motion cases like “Mobile and Calendar”.

This paper is to propose an adaptive motion estimation algorithm to get good tradeoff between coding efficiency and computation complexity. In section 2, we will give a review on the UMHS algorithm. We develop a new highly center-biased algorithm aiming at low motion sequence in section 3. The two algorithms will be combined together with an adaptive way in section 4. Final experiment result and analysis will be given in section 5, conclusions in section 6.

2. REVIEW OF UMHS ALGORITHM

The UMHS algorithm mainly includes four steps with different kinds of search patterns as Figure 2 shows.

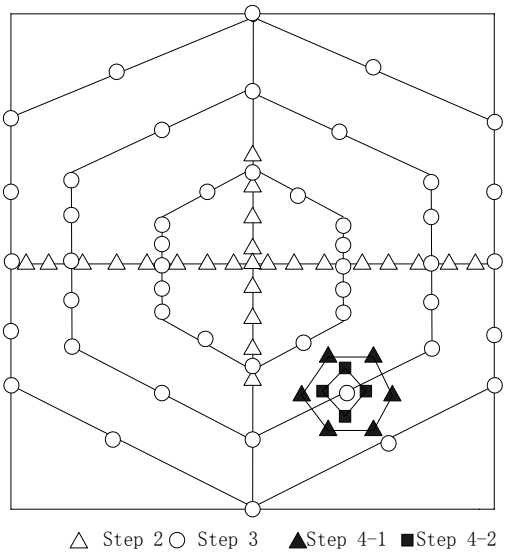

Figure 2 Search process of UMHS algorithm

Step-1: Initial search point prediction: Three candidates are checked, the median predicted MV, the MV of the larger blocks that contains the current block, the MV (0,0), the winner is the initial search point.

Step-2: Unsymmetrical cross search: an unsymmetrical cross search will be performed in this step, the center of the cross is the winner of step 1. Suppose the search range of motion estimation is W. then the length of the horizontal arm is W and the vertical one is W/2.

Step-3: Multi-hexagon-grid search: a multi-hexagon-grid search strategy is used in this step. The grid pattern is a 16-point hexagon. the grids’ scale are extended from 1 to W/4.

Step-4: Hexagon based Search: This step is to refine the search result of previous steps, firstly a 6-point hexagon search pattern is used, then a 4-point diamond search pattern is used.

The details of this algorithm can be referred to [4]. We can see that the search accuracy is the first consideration when this algorithm is designed. To solve the complexity of motion field, it has taken a uniformly search to iterate the search window in step 2 and step3.

3. CENTER-BIASED DIAMOND ALGORITHM

After all, in most cases the motion field is smooth and gentle, thus the MV prediction can be considered relatively accurate. So we have developed a highly center-biased and simple algorithm CBDS (center-biased diamond search) as Figure 3 shows.

Step-1: Initial search point prediction. This step is the same to that of UMHS.

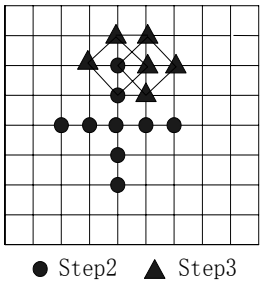

Figure 3 Search process of the CBDS algorithm

Step-2: Cross-center biased search. It is well known that the MV distribution is cross-center biased, so we take a cross-center search around the initial point and the arm length is 2, so only eight points are searched in this step. If the center point is the winner, the whole search stops, else it will go to step 3.

Step-3: A simple diamond-based search will be performed in this step. We choose diamond search pattern because it is compact than other patterns. The search will stop if the center of the diamond is the winner, else a new diamond will be searched with the winner as the center.

4. HYBRID ALGORITHM WITH ADAPTIVE COMPLEXITY

To illustrate the performance of the above two algorithms, we compare them to the FS (full search) algorithm in two typical sequences “Stefan”(strong

motion) and "Mobile and Calendar"(low motion). They are both 300 frame CIF sequences with the frame rate of 30f/s. The experiment is implemented in the reference software JM 7.3 of H.264, only the first frames is I frame and 2 B frames are inserted into two I/P frames, the search range is ± 16, RD optimization and Hardmard transform are used, 2 reference frames and all prediction modes are selected in the experiment. QP is set to be 32.

The computation complexity can not be simply evaluated by the count of search points, it is because that the block matching size in H.264 is variable and the partial SAD (sum of absolute difference) computation is used, the computation of a SAD is early terminated when the partially accumulated SAD has exceeded the minimum known SAD. So we use the count of AD (absolute difference) operations as the evaluation of a ME algorithm's complexity. An AD operation includes the subtracting between one pixel in the current block and corresponding pixel in the candidate block, getting the absolute value of difference, adding the absolute value to the SAD.

The following table gives the experimental result:

Table 1 Comparison of FS, UMHS and CBDS

	Algorithm	PSNR	Bitrate (Kb/S)	AD counts ($\times 10^9$)
Stefan	FS	31.68	667.63	127.95
	UMHS	31.65	669.23	11.21
	CBDS	31.56	825.08	4.17
Mobile	FS	29.97	484.19	110.34
	UMHS	29.94	481.09	10.21
	CBDS	29.93	481.04	3.29

From the above table, we can see that even in the sequence "Stefan" with strong motion, the conservative search strategy of UMHS succeeds in keeping similar rate-distortion property as the full search, while the CBDS loses badly, the bitrate increment is unacceptable. But the CBDS works well in the sequence "Mobile and Calendar", getting nearly the same coding efficiency with only 30% complexity of the UMHS.

The above test results show that the adaptive strategy can be considered as a proper solution. Now we predict the MV according to the MVs of the adjacent blocks. This is based on the assumption that the motion field is continuous. If the motion field here is smooth, then the MV of adjacent blocks should be similar and the real MV will be close to the prediction. But when the motion filed is complex the real MV will be far from the prediction. So we can regard the MVD (motion vector difference) between the real MV and the predicted MV as the indication of motion field's complexity. If the current block's neighboring blocks have large (small) MVD, the motion field here should be complex (smooth). To achieve the goal of spending the computation at where it is needed, we propose a hybrid algorithm incorporating the high complexity UMHS algorithm and low complexity CBDS algorithm together. A decision should be made before performing ME for a block.

When we design the adaptive ME algorithm for H.264, some clues for the decision should be noticed:

Firstly, there are seven block modes in H.264, Table 2 shows that the 16x16 block mode takes a dominant percentage among all the modes. So it is reasonable to spend more computation on the 16x16 block modes.

Table 2 Percentage of different block types

	16x16	16x8	8x16	8x8 8x4 4x8 4x4	Intra
Stefan	65%	8%	6%	13%	8%
Mobile	72%	6%	7%	14%	1%

Secondly, Table 3 shows the mean value of Euclidean distance between final MV and the predicted MV for each block type, we can see that the mean value is greater when the block size is larger, it can be explained by that the MV prediction is more accurate for smaller blocks. This indicates that more computation should be spent for larger block modes.

Table 3 Mean MVD distance of different block types

	16x16	16x8	8x16	8x8 8x4 4x8 4x4
Stefan	4.66	4.02	4.00	2.50
Mobile	1.22	1.25	1.25	1.00

Finally, H.264 has adopted intra modes for the P(B) frames, this mode is used when the ME fails to find good matches in the reference frame. If one neighbor of the current block is intra coded, it indicates that the motion here is not easy to predict and greater computation cost is needed.

Based on the above analysis, the details of the proposed algorithm can be depicted as follows:

If the current macroblock is on the top or left edge of the image, the UMHS is selected for the motion estimation of all the blocks in the macroblock.

For those blocks which are not on the boundary, we check their top, left, top-left neighboring blocks' MVD. If one of these neighbors is intra coded, the current block will take UMHS, else there are three pairs of neighboring MVD, let them be (Topx, Topy), (Leftx, Lefty), (Topleftx, Toplefty) and suppose:

$$T = \max(\text{abs}(Topx), \text{abs}(Topy))$$

L = max (abs(Leftx), abs(Lefty))

TL = max (abs(Topleftx), abs(Toplefty))

MaxMvd = max (T, L, TL)

(Note: abs(x) is the absolute value of x)

If MaxMvd is greater than a threshold Ti, the current block will take UMHS, else it will take CBDS. Note that the threshold is different to different block modes, there are totally 3 thresholds, T1 is for 16x16 block, T2 is for 16x8 and 8x16 block, T3 is for 8x8,8x4,4x8 and 4x4 block. And we have T1>T2>T3.

For a bi-direction predicted block, the forward/backward MVD of its neighbors is respectively used when performing forward/backward ME.

5. EXPERIMENTAL RESULTS AND ANALYSIS

To fully test the performance of our algorithm, besides the "Stefan" and "Mobile and Calendar", we add "foreman" (QCIF 300 frames) "Coastguard" (CIF 300 frames) and "Flower Garden" (CIF 240 frames), the frame rate of the all sequences are 30 frames/s. The experiment is done under the same condition in section 4. The MVD threshold T1, T2, T3 is heuristically chosen as 16, 32, 64 (in the unit of 1/4 pel accuracy).

Table 4 PSNR Comparison

	FS	UMHS	Proposed
Stefan	31.68	31.65	31.64
Coastguard	31.23	31.20	31.19
Foreman	32.55	32.50	32.50
Flower	30.42	30.40	30.38
Mobile	29.97	29.94	29.92

Table 5 Bit Rate Comparison (Kb/s)

	FS	UMHS	Proposed
Stefan	667.63	669.23	675.30
Coastguard	390.04	387.15	385.54
Foreman	59.21	59.30	59.93
Flower	638.09	638.99	642.36
Mobile	484.19	481.09	480.27

Table 6 AD Count Comparison ($\times 10^9$)

	FS	UMHS	Proposed
Stefan	127.95	11.21	7.29
Coastguard	124.70	11.53	5.43
Foreman	28.39	3.02	1.64
Flower	77.67	6.97	3.13
Mobile	110.34	10.21	4.15

We can see that in all the sequences the rate-distortion property of the proposed algorithm is nearly the same to the UMHS and FS, but the complexity is reduced according to the motion field of the video sequences, comparing to UMHS, the speeding up ratio is 50% in "Stefan", for the three sequences with moderate motion, the speeding up ratio is nearly 100%, and the proposed algorithm is more than two times faster than UMHS in the "Mobile and Calendar".

Furthermore, since a video codec may run on different platforms with different computation power, a computation-scalable strategy is desirable. The set up of threshold Ti in the proposed algorithm can easily satisfy this, more blocks will take the UMHS/CBDS as we reduce/increase T, making the complexity to be controllable.

6. CONCLUSIONS

In this paper, we firstly review the UMHS, which can maintain coding efficiency in video sequences with strong motions. Then we propose a center-biased hexagon based algorithm, it has low complexity but loses badly in video sequences with strong motions. To give a reasonable computation allocation, we propose the strategy to select the two algorithms adaptively according to the MVD of the current block's neighbors. The decision criterion is designed based on the character of variable block size ME of H.264. Experimental results show that this strategy can give a better computation-performance tradeoff. Furthermore, we can simply make the complexity scalable by adjusting the threshold parameter.

7. REFERENCES

[1] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438–442, Aug. 1994.

[2] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 369–377, Aug. 1998.

[3]Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Patten for Fast Block Motion Estimation". IEEE Trans, on CSVT, pp. 349-355, Vol.12, No.5, May 2002

[4]Zhibo Chen, Zhou Peng, Yun He "Fast integer pel and fractional pel motion estimation for JVT". JVT-F017 December 2002.

[5]"Draft ITU-T Recommendation and Final Draft International Standard for Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)", ITU-T SG16 Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG), JVT-G050r1, May 2003